Open Source Components, Standards Conformance, and UCD: Building Blocks for Successfully Managing and Enhancing an Established Digital Archive

Kathleen Murray and Mark Phillips; University of North Texas Libraries; Denton, Texas, USA

Abstract

The Portal to Texas History[SM] is a gateway to cultural heritage collections from Texas libraries, museums, archives, historical societies, and private collections. From its initial release in 2004, the Portal's unique visitors had grown from 1,000 per month to over 20,000 per month. The user interface had become dated and the underlying digital asset management system (DAMS) did not readily support implementation of new functionality. The IOGENE project at the University of North Texas Libraries involved family history researchers, a major user group of archives, in a user-centered application development project to redesign the Portal's interface. At the outset of the project, an application development model was created to guide three teams: system development, interface design, and user studies. The legacy DAMS was replaced with an infrastructure and framework of open source components. Specifications and standard practices in critical areas were established. The Portal's newly minted interface and infrastructure debuted in two public releases in 2009. Subsequent to each release, usability tests were conducted and at the conclusion of the project, experiences and accomplishments were reviewed by the project teams. This review informed a revised application development model that may be of value and interest to the both user support staffs and technical organizations at other archives.

Background

The University of North Texas (UNT) received a National Leadership Grant (LG-06-07-0040-07) from the Institute of Museum and Library Services for a two-year study to redesign the interface to the Portal to Texas History[SM], a digital library program at the UNT Libraries. In collaboration with over 100 content partners, the Portal is a gateway to cultural heritage collections from Texas libraries, museums, archives, historical societies, and private collections. The Portal contains primary source materials, including maps, books, newspapers, manuscripts, diaries, photographs, and letters, which are of interest to the family history researchers who participated in this project.

Since its initial release in 2004, the user interface to the Portal had become dated and constraints in the underlying technical infrastructure of the digital library impeded implementation of new functionality. Development was protracted and time-consuming and solutions did not scale well. Additionally, the number of unique visitors to the Portal had grown from 1,000 per month in 2004 to over 20,000 per month by 2008. This welcome growth was accompanied by operational and management challenges, which impacted the Portal's content partners, users, and other stakeholders. In short, development was not keeping pace either with desired features and enhancements or with commonly used and expected Web functionality.

Goals & Objectives

Two goals of the project were to overcome the constraints imposed by the legacy system and to create a user-centered application development model for interface development. In support of these goals, the project's objectives were:

- Implementation of a rapid development framework within the Digital Projects Unit of the University of North Texas Libraries, where development and design support for the Portal to Texas History reside.
- Creation of a model for the application of an iterative user-centered design process that digital libraries composed of cultural heritage collections could implement to improve the usability and effectiveness of their libraries and archives for targeted user groups.
- Demonstration of the iterative user-centered design model to create a new user interface, optimized for family history researchers, to the Portal to Texas History[SM].

Rapid Development Framework

A rapid development framework is configured using interchangeable and robust modules, components, and tools, which are both highly cohesive and loosely coupled. In building the framework, deliberate attention is paid to selecting components that are supported by active user communities and audited by a large base of users. Conversely, components and tools that are developed for a niche community are avoided. Components at each level within the framework are highly scalable, allowing for distribution of costs across the framework as increased capacity is needed.

Rapid application development strives to design and deliver applications within a relatively short timeframe (e.g., 30-90 days) [1]. Solutions strive to be simple and straightforward as well as portable and standards-based. The goal is to compress development into as few phases as possible, resulting in more user-responsive application development. To achieve this goal, functional requirements are first identified and generally remain unchanged during the application development process. Subsequent to the identification of functional requirements, developers use prototyping tools to create functional designs, which are revised in response to user feedback in an iterative process. Digital library development is well-served by an iterative, process-oriented approach [2] and within a rapid development

framework this continues until a final prototype is established. Development of the fully functioning application then occurs. Testing is generally done concurrently with development, once again optimizing development time.

Digital Asset Management Systems

Collections of cultural heritage materials, such as those comprising the Portal to Texas History[SM], include a range of digital resources or assets, such as photographs, maps, diaries, newspapers, and books. The digital asset management systems (DAMS) commonly implemented for these collections have historically offered little user interface design flexibility to developers [3]. Rather, DAMS have concentrated on providing tools and workflows to assist providers and creators of digital assets, with an added focus on description of the system's digital assets. Anecdotal evidence attests to the difficulties user interface developers encounter when they seek to change the user experience of most DAMS. Open source solutions often offer increased flexibility to tailor development efforts [4] to a particular design effort.

Application Development Model

At the start of the IOGENE project, an application development model that incorporated user-centered design methods was drafted to guide the project (Figure 1). Throughout the project, family history researchers were incorporated into the process, beginning with an assessment of their requirements and concluding with their participation in usability testing.

The working model included three teams within the Information Technology Services (ITS) department at the UNT Libraries: System Development to configure, code, and test infrastructure components; Interface Design to create and code the user interface components; and User Studies to assess user needs, develop functional requirements, and conduct quality and usability testing. The model anticipated that the UNT Libraries' legacy DAMS would be replaced by a rapid development framework at the outset of the project. This enabled separation of the interface design and system development functions and allowed the project to include the unique expertise of the Libraries' user interface designers in the project.

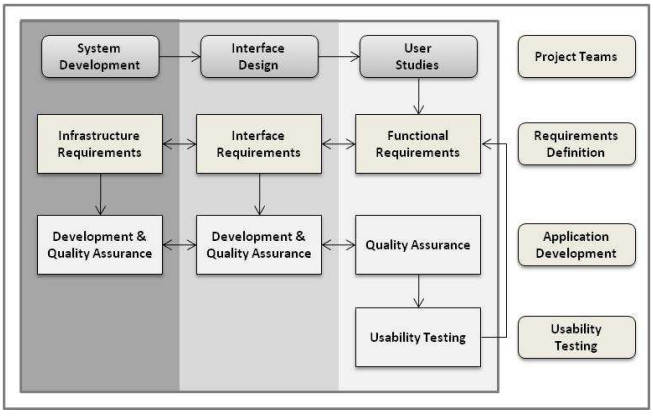

Figure 1. *Draft Application Development Model*

The model envisioned three iterative phases: Requirements Definition, Application Development, and Usability Testing. Quality assurance activities were emphasized and the interactions and interfaces among the teams were identified. The arrow from Usability Testing to Functional Requirements in Figure 1 indicates a key feedback loop in the model, that is, usability test findings became input for subsequent requirements.

Requirements Definition

Infrastructure Requirements

Although not all the configuration details were specified, the infrastructure requirements and components for the rapid development framework were largely known entities that had to be implemented at project start-up. Figure 2 illustrates the external and internal systems comprising the Portal's core infrastructure components.

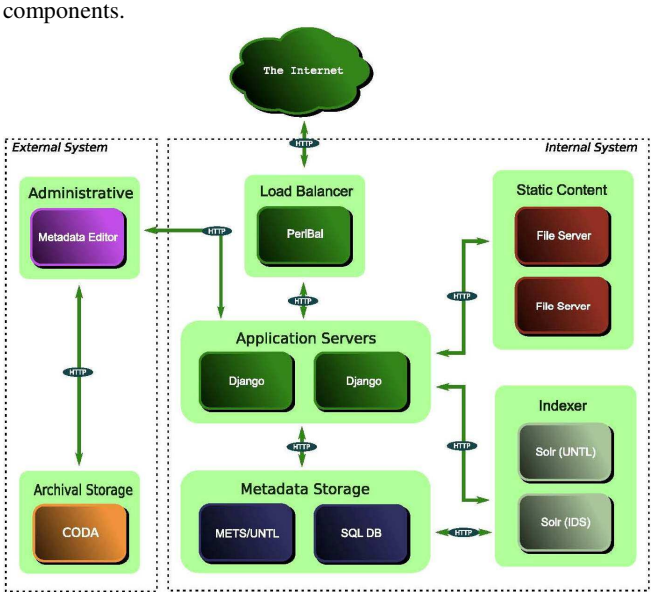

Figure 2*. Core Infrastructure Components*

The external system includes the Metadata Editor, an Administrative component supporting metadata creation and editing for digital objects, and an Archival Storage component locally referred to as CODA (Complex Object Digital Archive). Long-term stewardship and curation of digital content is handled within the CODA system. The internal system has five core components, which communicate via HTTP.

Internal System Components

1.	Static content	File servers for digital objects (image files, OCR-text, bounding box information)
2.	Indexer	Solr search servers with both object level and page level indexes. http://lucene.apache.org/solr/
3.	Metadata Storage	Metadata Encoding and Transmission Standard (METS), UNT Libraries (UNTL) metadata, and MySQL database http://www.mysql.com/
4.	Application Servers	Django framework for website design and development http://djangoprojects.com/
5.	Load Balancer	PerlBal, a perl HTTP load balancer http://www.danga.com/perlbal/

When possible, open source components and tools were selected. The following components were included in the framework:

Open Source Components & Tools

Apache	Server Software http://www.apache.org/
jQuery	JavaScript Library http://jquery.com/
Memcached	Distributed memory object caching system http://www.danga.com/memcached/
Python	Programming Language http://www.python.org/
mod_python	Apache module that embeds the Python interpreter within the server http://www.modpython.org/
Subversion	Version Control System http://subversion.apache.org/
Ubuntu	Operating System http://www.ubuntu.com/
Trac	Issue tracking system for software development projects http://trac.edgewall.org/

In addition to the core components and tools, infrastructure requirements included specifications and practices that enabled implementation and testing of prototype technologies and standards prior to final implementation. The specifications were:

1. Persistent Identifiers
   Archival Resource Keys (ARKs) [5] were implemented as part of the persistent identifier strategy. Digital objects within the system were mapped to URLs, with ARKs playing a key role in providing logical, hack-able and bookmark-able identifiers for the system.
2. Digital Object Manifestations Model
   A model for defining a digital object entity and for defining digital objects in a consistent manner was created. Based on existing and future content, the object model allowed the development team to create standardized tools for reading and writing digital objects. METS [6] was used as a serialization format for this object model throughout the system.
3. Metadata Scheme
   The UNT Libraries uses a locally qualified Dublin Core metadata format (UNTL) for all digital collections. The UNTL input guidelines and formatting rules were updated to reflect the new data model and the metadata scheme introduced during the project.

Requirements for a new application ("Edit") to facilitate modification of records in the new system were created. The Django framework and application interfaces to system components are illustrated in Figure 3.

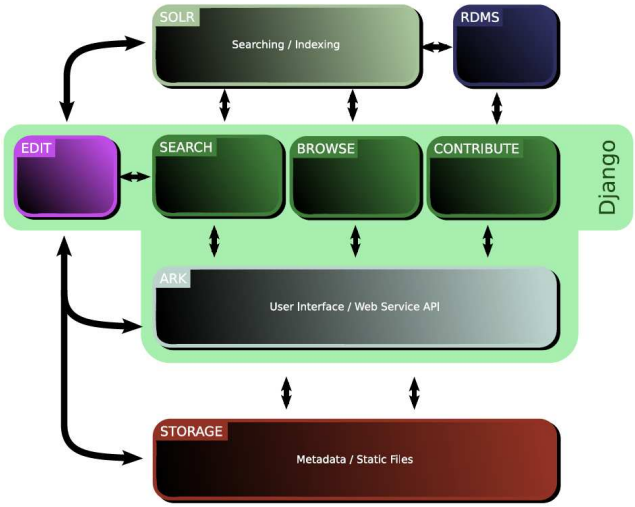

***Figure 3**. Application Interfaces to System Components*

Interface Requirements

Five design possibilities for the Portal's user interface were created. Additionally, a survey regarding design characteristics and preferences was completed by members of the Digital Projects Unit. The results were used to guide a group discussion among these internal stakeholders regarding priorities for the Portal's redesign in three areas:

1. Overall design
2. Functionality
3. Information architecture/content

Paper prototypes illustrating navigation structures and page layouts were developed (Figure 4). The design of these prototypes was informed by the results of the initial usability testing, by an evaluation of the content structure of current Portal, and by the functional requirements generated from focus group findings.

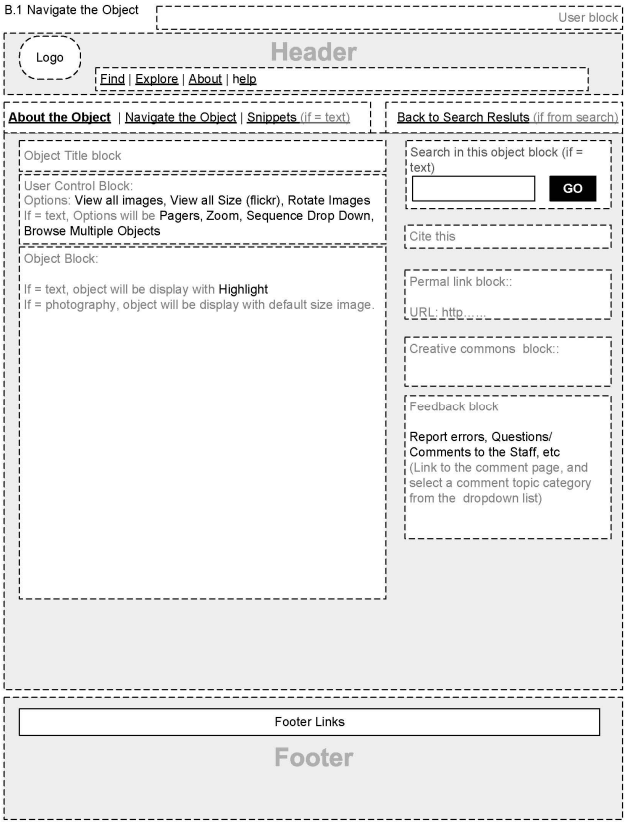

***Figure 4**. Paper Prototype for an Object Navigation Page*

After review and refinement of the paper prototypes, HTML mockups were created for both object metadata pages and object navigation pages. The purpose of mockups was to visualize all interface related requirements and to expose potential issues. These pages were informed by complex requirements and were an integral part of the display of every digital object in the Portal. Hence, great care was taken to ensure that the features to be implemented would serve users' needs. Various storyline walkthroughs were conducted and revisions were made as needed.

Functional Requirements

System Development and Interface Design teams generated questions, ideas, and prototypes regarding possible features and functional enhancements to the existing Portal. The User Studies team used these to develop protocols for focus group discussions with members of genealogical societies (*N*=19). The findings from these discussions, as well as an analysis of the Portal's historical log of user-submitted comments, and results from initial usability tests informed the functional requirements drafted by the User Studies team. These requirements were refined by members of the three teams and classified for implementation in two releases. Interface requirements and infrastructure requirements needed to support the functional requirements were also identified.

Application Development

After determining the requirements for the new system infrastructure, components, tools, and interfaces, conversion from the legacy system to the new rapid development framework commenced. This involved:

- Implementation of a new document scheme for Solr documents representing a digital object:
  - Object Level Solr representation
  - Page Level Solr representation
- Implementation of a conversion script to migrate from the format in the legacy TKL system developed by IndexData to the new METS format
- Conversion of the in-house digital object indexer (IREX) to a new version that supports both METS and the new UNTL metadata scheme
- Implementation of a system for mapping Archival Resource Keys (ARKs) to the underlying digital objects stored as METS files, including authentication and access restrictions
- Specification of the branding application

Servers were configured for static media (image files, OCR-text, and bounding box information) and metadata (METS and UNTL). Initial conversion of the Portal's content was completed for use in the prototype of the rapid development framework. A development environment was created for multiple developers, each working with different components of the framework.

During the development process, the System Development team worked closely with the Interface Design team to implement components. They created workflows to support their separate but complementary design and development work. As the Interface Design team developed the user interface, business logic was added as needed by the System Development team to provide access to data required for the user interface.

The System Development team created and tested the "Edit" application, as well as three ingest tools to enable new content additions to the system. Interface designers implemented and customized the CSS framework for the web interface and developed JavaScripts to support functional requirements. Interface designs were implemented for these template sets:

- Portal Home
- Basic Search
- Advanced Search
- Search results
- About the Object
- View/Read the Object
- Explore: Collection, Partner, Location, Subject, Date, and Type
- Documentation: Help, FAQ, and Guides
- About

Quality Assurance

Subsequent to completion of a Release 1 beta system, a structured Quality Assurance (QA) test script was created by the User Studies team. Members of the Information Technology Services (ITS) staff within the University Libraries completed the scripted tasks. The User Studies team analyzed the test results and the findings informed a set of design and development tasks that resulted in a revised beta site for Release 1. A second round of QA

testing for Release 1 included ITS staff and practicing genealogists. Once again, test feedback informed a set of design and development tasks that were completed prior to the public launch of Release 1. Testers reported an overall success rating of 84% for the 46 tasks in the first QA test, and 90% success for the 37 tasks in the second test.

Prior to the public launch of Release 2, QA testing was again conducted with ITS staff. With the exception of one task, 73% or more of the testers indicated they successfully completed each of the 17 tasks, although some noted issues and problems with some tasks. As before, test feedback informed system design changes.

Usability Testing

Release 1 of the redesigned Portal was public launched in June 2009 and Release 2 in October 2009. Subsequent to each release, the User Studies team conducted usability tests with members of genealogical societies (*N*=7 for Release 1; *N*=6 for Release 2). Usability testing of Release 1 of the Portal interface identified areas for revisions to the interface, primarily in regard to secondary navigation features. Illustrative video clips were created by the User Studies team to highlight user behaviors and issues for the design and development teams. Revisions to the interface were subsequently created.

Usability test results following the public launch of Release 2 were positive. The average completion scores for only three of 42 tasks resulted in completion failures. These tasks were among those that tested users' ability to locate secondary navigation features.

Project Evaluation

Subsequent to the launch of Release 2, the project teams engaged in a project review. Group discussions sought to identify what had worked well and what had not. Findings from this evaluation informed a revised application development model (Figure 5).

Stakeholder Involvement

Stakeholders, both within and outside the Libraries, need to be represented and/or included in the requirements review and in establishing development priorities. Features that relate to content partners need to have their input either directly or via the program manager(s). Internal stakeholders, including program managers, need to be cognizant of system development plans.

Prototyping Design Requirements

This project conducted a user assessment that informed a set of functional requirements for the Portal's redesign. In the future, clarification of user requirements might be better achieved with continuing user involvement through prototyping of the user interface. This would involve interface designers creating paper-based or online mock-ups of user workflows and conducting usability tests of the workflows. The findings would result in a set of final design requirements.

Impact of Technology Changes

It is important to anticipate and plan for technology changes in infrastructure components, including operating system and component upgrades. The further up the technology infrastructure chain, the more frequently changes occur. For example, the web framework (CSS framework) changed three times over the course of the project. Longer-term projects would likely experience more changes.

Design Strategies

Given finite resources, it is prudent for a development organization to follow the design leadership of industry leaders who invest heavily in usability testing. Following their leadership, in terms of features and design, effectively leverages the results of that investment in testing.

De-coupling interface design from development of the underlying system is a key to readily making changes to the user interface, including required upgrades. For example, design templates could be changed independently and without impacting the underlying system, and vice versa.

Estimating Time and Resources

Adequate resources, in terms of people and time, need to be identified for infrastructure implementations and system migrations. The scope, challenges, and learning curves for implementation of the infrastructure in support of the rapid development framework were not adequately estimated. As a consequence the amount of time and resources required were not adequately estimated and completion dates were delayed.

Framework

Implementation Challenges

From a system perspective, roughly half of the project involved implementing the system framework. The framework had to be in place prior to beginning development of the user interface. A great deal more time than anticipated was needed to write conversion code to migrate from the format in the legacy TKL system to the new METS format and to create and implement new backend workflows for moving digital objects in and out of the system.

Management of Application Development

The Subversion system provided a running log of all changes and facilitated ticket assignments for development and refinement. However, integrating a second programmer into the new system was time-consuming and adding two additional programmers to the Django framework was challenging. With three developers, it is critical to understand who has access to and who needs to know about changes. With more developers, workflows and additional rules would need to be enforced.

Benefits

The framework enabled (a) separation of user interface design from development of the backend system and (b) specialization of team members in technology areas. The use of the Django template system for user interface development enabled a faster and more scalable development environment. Components, such as Django, allowed for development of reusable applications and open source tools gave developers beneficial access to a large external community of developers.

Quality Assurance Testing (QA)

Engaging users in quality assurance testing prior to public release of the application resulted in valuable feedback to both the User Interface and System Development teams. Users brought issues to light that internal tests had not uncovered, for example, Django and Python testing did not uncover character problems, such as problems with diacritics or ampersands. It would be advantageous to include Portal partners and library stakeholder groups in QA testing to foster a sense of the system as "theirs".

Application Development Model

In light of what we learned during the project, the initial draft model for application development was revised (Figure 5). The revised model includes a fourth project team, Program Management, who represent content partners, external stakeholders, funding agencies, and a particular digital library program, such as the Portal to Texas History[SM].

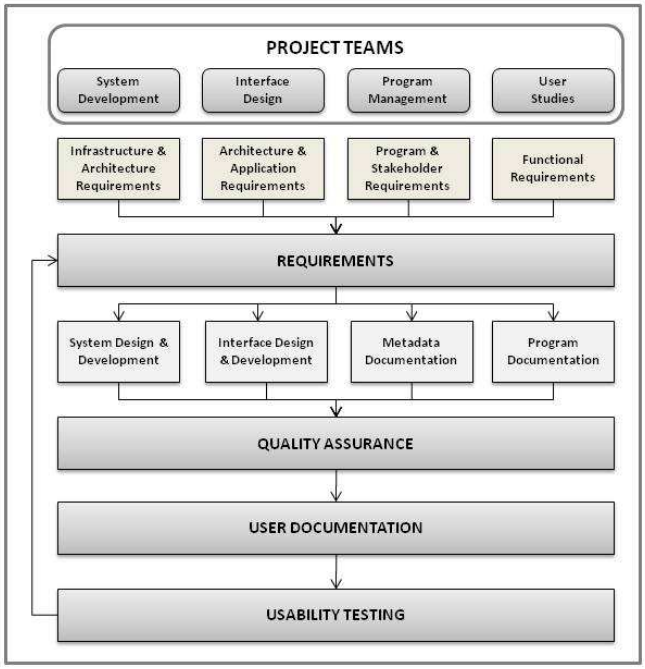

Figure 5. *Revised Application Development Model*

Program managers are aware of the needs of external stakeholders and end users. They communicate those needs and translate them into requirements for digital library operations and services. They are knowledgeable go-betweens among stakeholder groups, library administrators, funding agencies, and the digital library support staff.

The new process model also makes explicit the activities of metadata documentation and program documentation in the application development process. Likewise, the creation of user documentation is now clearly situated in the model to ensure that applications are deployed with documentation in place and that usability testing can encompass user documentation.

Closing

This project created a model for digital library application development informed by user-centered design methods and supported by a rapid development framework. At the onset, family history researchers were involved in the application design process as participants in focus groups structured to gain an understanding of their information needs. The findings informed a set of functional requirements for redesigning the existing interface to the Portal to Texas History[SM], a digital library program at the University of North Texas Libraries. Family history researchers were also involved in usability testing of two public releases of the redesigned Portal interface.

Substantial amounts of time and effort were invested in the specification and implementation of components for the rapid development framework. This experience reinforced the importance of accurately estimating the time and resources required to implement backend infrastructure components. That said, the investment achieved beneficial results: the constraints of the legacy digital asset management system in terms of new feature implementation were alleviated. Interface designers and backend system developers are able to work independently, yet in concert, to optimize the application development process. Lastly, the new framework has proved robust at handling an ever-increasing number of visitors, as attested by the 59% increase in the number of Portal visits per month from June 2009 to January 2010.

References

[1] J.D. Fernandez, M.A. Martinez-Prieto, P. de la Fuente, J. Vegas, and J. Adiego, "Agile DL: Building a DELOS-Conformed Digital Library Using Agile Software Development", Proc. ECDL, pg. 398. (2008).

[2] A. Bishop, N. VanHouse, and B. Buttenfield, Eds., Digital Library Use: Social Practice in Design and Evaluation (The MIT Press, Cambridge, MA, 2003) pg. 6.

[3] D. Salo, "Innkeeper at the Roach Motel," Library Trends, 57, 2 (2008).

[4] R. Uzwyshyn, "Repurposing Open Source Software for Agile Digital Image Library Development: The University of West Florida Libraries Model," D-Lib Magazine, 14, 9/10 (2008).

[5] California Digital Library, Univ. of California, "ARK: Archival Resource Key," (2008), accessed April 16, 2010 at https://confluence.ucop.edu/display/Curation/ARK

[6] Library of Congress, "METS: Metadata Encoding and Transmission Standard," (2010), accessed April 16, 2010 at http://www.loc.gov/standards/mets/

Author Biography

Kathleen Murray received her PhD in information science from the University of North Texas (UNT, 2000). As a postdoctoral research associate, her work focuses on user studies in the areas of digital libraries and web archives. Mark Phillips (MLS, 2004) is head of the Digital Projects Unit at UNT. He manages the UNT Libraries' software development projects, including infrastructure design, software development, and digital content creation.